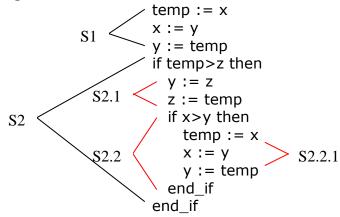
## **Software Testing and Verification**

## **Problem Set 6: Predicate Transforms – Solution Notes**

- 1. a. Would.  $[wp(s, y=z) = z>-5] => [\{t=5 \land -5 < z < 0\} \ s \ \{y=z\} \ strongly] => [\{t=5 \land z < 0\} \ s \ \{y=z+1 \land t=z\} \ is FALSE] since we know s will halt for at least four initial (integer) values of <math>z < 0$  with post-condition  $(y=z+1 \land t=z)$  being false.
  - b. Would not.  $[wlp(s, y=z) = z>-5] => [\{t=5 \land z>-5\} s \{y=z\}]$  weakly. But this does **not** imply that  $\{t=5 \land z<0\}$  s  $\{y=z+1 \land t=z\}$  is FALSE since s may not terminate for -5< z<0.
  - c. Would. In order for  $\{t=5 \land z<0\}$  s  $\{y=z+1 \land t=z\}$  to hold weakly, it is necessary that  $(t=5 \land z<0) => \text{wlp}(s, y=z+1 \land t=z)$ . But if  $(t=5 \land z<0) \neq> \text{wlp}(s, y=z+1 \lor t=z)$ , then it would also be the case that  $(t=5 \land z<0) \neq> \text{wlp}(s, y=z+1 \land t=z)$ , which is an even stronger condition.
  - d. Would not. ["sp(s, t=5  $\Lambda$  z>-5) is undefined"] implies that s does not terminate when t=5  $\Lambda$  z>-5 initially, which is consistent with the given assertion of **weak** correctness for -5<z<sub>0</sub><0. [sp(s, t=5  $\Lambda$  z≤-5) => (y=z+1  $\Lambda$  t=z)] implies that the specified post-condition, (y=z+1  $\Lambda$  t=z), will hold on termination for z<sub>0</sub>≤-5, which is also consistent with the given assertion.
- 2. For program S:



we need to determine:

$$wp(S, x \le y < z) = wp(S1; S2, x \le y < z)$$

$$= wp(S1, wp(S2, x \le y < z))$$

$$= wp(S1, wp(if temp>z then S2.1; S2.2, x \le y < z))$$

$$= wp(S1, (temp>z \land wp(S2.1; S2.2, x \le y < z)) \lor$$

$$(temp \le z \land x \le y < z))$$

- 3. ({P} s {Q}) does not imply (P=>wp(s,Q)) because weak correctness (which does NOT require that s terminate) does not imply strong correctness (which DOES require that s terminate).
- 4. a. wlp rule for while-do statements:

## $wlp(while b do S, Q) \equiv wp(while b do S, Q) V \neg wp(while b do S, true)$

i. determining wp(while b do S, Q):

H<sub>0</sub>: 
$$J=Y \land Z=XY$$

H<sub>1</sub>:  $J<>Y \land wp(s, J=Y \land Z=XY)$ 
 $= J=Y-1 \land Z=X(Y-1)$ 

H<sub>2</sub>:  $J<>y \land wp(s, J=Y-1 \land Z=X(Y-1))$ 
 $= J=Y-2 \land Z=X(Y-2)$ 

H<sub>k</sub>:  $J=Y-k \land Z=X(Y-k)$ 

Therefore, H<sub>0</sub> V H<sub>1</sub> V H<sub>2</sub> V ... V H<sub>k</sub> V ... simplifies to:

$$(J=Y \land Z=XY) \lor (J$$

ii. determining wp(while b do S, true):

H<sub>k</sub>: J=Y-k

Therefore,  $H_0 V H_1 V H_2 V ... V H_k V ...$  simplifies to: **J** $\leq$ **Y** 

Thus, wlp(while b do S, Q)  $\equiv$  (J $\leq$ Y  $\wedge$  Z=XJ) V J>Y = **Z=XJ V J>Y** 

b. Obviously, the wlp is weaker than the given invariant, i.e.,

$$(Z=XJ) => (Z=XJ V J>Y)$$

This makes sense since the wlp is the **weakest** condition on the initial state of program S ensuring state Q on termination *if* S *terminates*. If J>Y initially, the program will obviously not terminate, implying weak correctness *whether* Z=XJ *holds initially or not*.

5. We wish to prove:  $(wlp(while b do s, Q) \land \sim b) => Q$ . The left hand side of the implication,  $(wp(while b do s, Q) \land \sim b)$ , is:

6. a. 
$$H_1 = wp(s, c \land Q)$$
  
 $H_2 = wp(s, \sim c \land H_1)$   
 $H_3 = wp(s, \sim c \land H_2)$   
 $H_k = wp(s, \sim c \land H_{k-1})$ 

b. 
$$H_1 = wp(s, c \land Q) = wp(s, y=0 \land x=17)$$
  
  $= y-1=0 \land x+1=17$   
  $= y=1 \land x=16$   
 $H_2 = wp(s, \sim c \land H_1) = wp(s, y<>0 \land y=1 \land x=16)$   
  $= y=2 \land x=15$   
 $H_3 = y=3 \land x=14$   
 $H_k = y=k \land x=(17-k)$   
 $wp \text{ (in closed form)} = y>0 \land x=17-y$